Predictive Analysis of Errors During Robot-Mediated Gamified Training

Nihal Ezgi Yücetürk[1], Sevil Demir[3], Zeynep Özdemir[3],
Irina Bejan[1], Nevena Drešević[1], Marija Katanić[1], Pierre Dillenbourg[1], Aysun Soysal[3], Arzu Guneysu Ozgur[1,2]

***Abstract*— This paper presents our approach to predicting future error-related events in a robot-mediated gamified physical training activity for stroke patients. The ability to predict future error under such conditions suggests the existence of distinguishable features and separated class characteristics between the casual gameplay state and error prune state in the data. Identifying such features provides valuable insight to creating individually tailored, adaptive games as well as possible ways to increase rehabilitation success by patients. Considering the time-series nature of sensory data created by motor actions of patients we employed a predictive analysis strategy on carefully engineered features of sequenced data. We split the data into fixed time windows and explored logistic regression models, decision trees, and recurrent neural networks to predict the likelihood of a patient making an error based on the features from the time window before the error. We achieved an 84.4% F1-score with a 0.76 ROC value in our best model for predicting motion accuracy related errors. Moreover, we computed the permutation importance of the features to explain which ones are more indicative of future errors.**

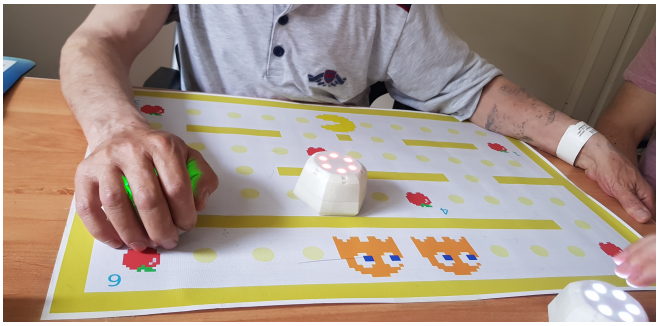

Fig. 1: An example gameplay of a stroke patient while manipulating a robot with the hand of the affected side on a paper maze with the theme of arcade Pacman game.While the patient collects red targets on the maze with yellow walls, another autonomous ghost robot chases the user.

I. Introduction

Among the characteristics that should be proposed by a training intervention, *adaptation* merits special attention. It is a crucial aspect for keeping the user engaged with the activity [1], while also being a prerequisite to ensure that the activity is tailored to multiple user-centered dimensions such as the user's capabilities, rehabilitation goals, and interests.

Previous studies on gamified and robotic therapy focus on monitoring the execution of the exercises and provide online in-game adaptations according to the patients' impairment level, the current in-game performance and progress, the exercise plan and goals specified by the therapist or the emotional state of the patient [2], [3]. Proposed strategies include modifying the gameplay through adjusting the game elements and the difficulty of the game and/or activating or modifying the assistance of a robotic device [4]. These adaptations can provide personalized interventions for individually tailored user-specific training to increase engagement and active participation by providing optimum challenge level for each patient [5].

In order to provide optimum challenge level, the current performance needs to be monitored and task parameters have to be adjusted throughout the training to create a state of flow for the patient [6], [7]. This optimal challenge level should be accomplished in a way that the challenges are not greatly higher than the skills of the user, which would lead to frustration or anxiety, and the user's skills are not greater than the proposed challenge level, which would lead to apathy or boredom [6], [7], [8].

In previous studies, real-time performance-based adaptation strategies, such as Bayesian methods [3] as well as ad-hoc methods, use the interaction or performance history of the player in the game and then modifies the difficulty level to match the skills of the patient. In this research effort, we propose the integration of future error probabilities into the adaptation strategies and as an initial step towards this goal, we introduce using different ML methods to do predictive analysis of future errors during gamified training. We believe that together with previously used metrics such as the performance history, a possible prediction of future errors is beneficial for the development of more personalized systems.

Although one of the main triggers for user frustration might be errors, previous research exploring the value of users' possibility of failure suggests that avoiding the mistakes is part of the challenge for the user. The joy of success is dependent upon the possibility of failure [6]. Since the failures have a crucial role in engagement, combining the previous failure information of the user along with the future probability of doing an error might be a valuable contribution to the design of more robust adaptive strategies. The addition of early error prediction to the previously suggested adaptive strategies, which rely on adapting the intervention according

This work is partially supported by NCCR Robotics, Switzerland and Digital Futures Research Center, Sweden

[1]Computer Human Interaction in Learning and Instruction Lab (CHILI), EPFL, Switzerland. name.surname@epfl.ch

[2]Division of Robotics, Perception, and Learning (RPL), KTH, Sweden. arzuo@kth.se

[3] Bakirkoy Prof. Mazhar Osman Research and Training Hospital for Psychiatric and Neurological Diseases, Turkey.

to different parameters (e.g. the performance, frustration level, gaze, brain signal, or muscle activity of the user), might result in a more optimized adaptation strategy.

Apart from determining *which aspect or element of the task to adapt* (e.g. robot speed, the difficulty of the game) and *which metric to adapt for* (e.g. performance or muscle activity of the user), in an online personalized system, it is also important *when and when not* to implement the corresponding adaptation (e.g. after an error or low performance, after a game failure or according to some thresholds), especially for assist as needed therapeutic interventions [4]. These assist as needed systems such as therapy robots and integrated neuro-muscular stimulation devices are used for the patients who cannot actively continue the physical therapy exercises for long due to weak muscle control or fatigue [4]/ Timing of the use of such technologies' aid is crucial for the therapy. Previous studies investigate the best timing for adaptive assistance for those patients through measuring error-related potential through brain signals [4]. The second possible advantage of early prediction of the errors in therapy might be using the error prediction for determining the timing of activating these assist-as-needed platforms in a combined system approach that integrates our proposed game with such systems. We believe that the addition of early error estimation might help to decrease the uncertainty of robot aiding time beforehand the event of the error. Therefore, apart from using error prediction information *as an additional metric in adaptive strategy* it can also be useful in *determining the timing of the adaptation* for assist as needed technologies.

In this work, we explore various machine learning methods to predict the probability of an error happening in the next time frames in the future gameplay. We used the data collected from 29 stroke patients during the gameplay of Tangible Pacman which is designed as an upper limb rehabilitation exercise [9] to provide easy to use and intuitive gamified rehabilitation intervention by using tangible robots and paper game spaces as can be seen in Fig. 1.

We compare the performances achieved by different models, including logistic regression, random forests, XGBoost, feed-forward neural networks, recurrent networks, and fine-tuned various problem hyperparameters, such as the window size, and the number of backward and forward time frames considered for predictive analysis of two types of errors. We further analyze the models to explain what features are more indicative of future errors, gaining more insights into the relationship between the motor performance of the patients and the errors.

II. Data Collection

A. Robot-Mediated Gamified Training Activity and Error Types

Gamification is defined as the use of game design elements in non-game contexts which is also integrated into therapy as a strategy to increase patient engagement [10]. It has been proposed that when designing gamified applications for rehabilitation, two game design principles are of particular importance: meaningful play and challenge [11]. In the light of these principles, Tangible Pacman was iteratively designed as a novel upper extremity rehabilitation game for mainly targeting shoulder and elbow exercise with tangible robots involving diverse patient groups such as stroke patients and children with hemiplegia, from 3 to 77 years old within different therapy environments [9].

Tangible Pacman consists of two to three Cellulo robots including one called Pacman, which is manipulated by the player to collect six target apples on a printed paper maze (see Figure **??**. Each Cellulo robot is a palm-sized, mouse-like object which has multiple sensors and through sub-mm accurate localization on the dot patterned paper it can measure the motion performance of the patients with sub-mm accuracy, please see [12], [13] for the detailed design of the robotic system and [9], [14] for the detailed game design.

The goal of the game is to collect the all apples as soon as possible while running away from the ghosts robots (which are autonomously chasing the Pacman robot through the gamespace and not crashing into the walls. If an autonomous ghost robot catches the Pacman, the player loses all previously collected apples, and the game restarts. This event is one of the error events in our predictive analysis and is called as *ghost catch error* throughout this paper. The game consists of several tunable game elements co-designed with stakeholders such as the number of ghosts, speed of the ghosts, haptic assistance (which is the direction based informative assistive feedback with a maximum force of 1 Newton applied upon crashing the wall, the robot moves towards the middle of the path and it does not actively move patient's hand) and wall crash penalty rule, for more detailed information please see [9]. During the data collection, the game configuration is adapted to the impairment level of the patients, and the difficulty of the game is increased over time to avoid frustration as well as boredom. Some example adaptations are as follows: for the patients who move slowly, the ghost speed and number of the ghosts are decreased, a penalty rule is implemented for the patients who are able to move fast, and a turn rule is introduced for the patients who are able to manage to rotate the robot.

Our second error type is related to the penalty rule. If the rule is on when the player crashes into a wall of the maze by crossing the wall borders, the last eaten apple is lost as a penalty for the crash. These wall crash events are more motion accuracy-related events and it is the second error type, named *border cross error*, we focused on in our predictive analysis work.

B. Dataset

The data set was collected from 29 mildly or moderately impaired stroke patients (27 to 76 years old with Fugl-Meyer Upper Extremity (FMA-UE) scores between 20 to 66, who are able to hold the robot with minimal assistance) with the approval of The Ethical Committee of Bakirkoy Prof. Mazhar Osman Research and Training Hospital for Psychiatric and Neurological Diseases. The data was composed of a total of 810 gameplays with adapted game configurations according

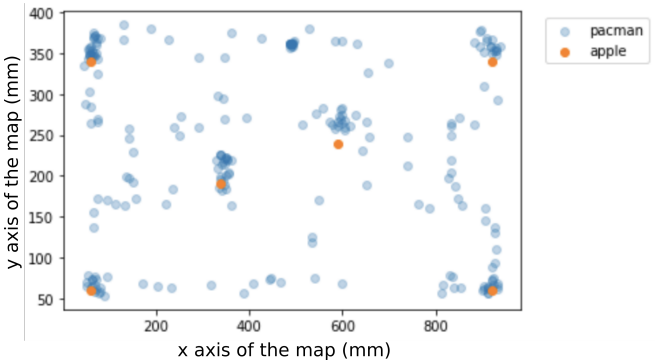

Fig. 2: Blue dots represent the positions of the player in the moment of crossing a border, while red ones are positions of the apples on the map.

to the impairment level of the patients. Three different maze maps of different sizes and one proper-sized map were used by each patient according to the ranges of motion of each patient.

During the game, the system continually logged the players' hand position (Pacman robot held by the patient) and ghosts' positions in sub-mm accuracy as the time-series data at a 10 Hz rate. Also, every moment of making an error was captured, which gave two different data sets based on the error type. Besides this real-time information, we received patient specific information, out of which we used the hand they play with (left or right). We preprocessed the data and crafted the features to build the model that predicts if a patient will make an error in the next x seconds of the game.

C. Data Preprocessing and Feature Engineering

Since stroke is related to different motion characteristics of motor performance, we computed several features by using position data of the hand of the patient which are among the highly adopted metrics in the kinematic assessment of upper limb movements in the literature [15] and mostly focusing on the speed, accuracy, and smoothness characteristics of the motion. Features are listed as follows with corresponding motion characteristics in parenthesis;

- Mean and max values of overall velocity (speed), acceleration (smoothness and accuracy), and jerk (smoothness) during the gameplay (v_mean, acc_mean, jerk_mean, v_max, acc_max, jerk_max) specific means of velocity and acceleration in x and y directions (vx_mean, vy_mean, accx_mean, accy_mean)
- Mean deviance from the middle of the path (deviance) (accuracy)
- The mean/max ratio of the overall and direction specific velocity (v_ratio, vx_ratio, vy_ratio) (smoothness)
- The total number of velocity peaks (number_v_peaks_per_time) (smoothness)
- Time to the maximum velocity peak (time_to_maxpeak) (smoothness)

Further feature engineering indicated how making the error by crossing the border was dependent on the closeness to the apples (see Fig. 2). This might be an indicator of the patients' difficulties in controlling the movement and maybe overshooting, primarily when their attention was devoted to collecting the apple. Due to such relations, we computed distance-related features such as mean distance to the center of the map (min_dist_to_center) and split the maps into regions, and added this as a feature called the sector of the map where the user is moving on the map during the play. We also used patients' affected hand sides (left and right) as a feature since it might affect the error locations on the map sectors. We standardized all the numerical features by subtracting the mean and dividing by the standard deviation of the feature (z-score) and fed this data into the machine learning models.

Since the two types of possible user errors might have different causes, we prioritized different features for each error to better train our models. For instance, when predicting if ghosts will catch the player, we considered the ghosts' positions in time and distance of Pacman to the nearest and the farthest ghost.

III. Methods

The first challenge was modeling the foretelling of in-game errors. Foretelling of an event based on some features poses a complex task, a prediction problem, which could be addressed by machine learning methods. Given the nature of the features that derived from the hand movements, our approach relied on predictive modeling on time series data. Such models use past and current data in time to reliably forecast trends and behaviors in the future. We stated an initial binary classification problem of whether an error will happen in a specific time window from now, considering a fixed number of windows in the past.

A. Imbalance of Error-Bearing and Error-Free Windows

For feature extraction and error prediction, a certain window size should be selected to divide the data. However, by decreasing the size of the time frame (e.g. from 10s to 3s), we were able to capture the granularity of the movements more, but given the comparatively small number of errors during the total gameplay, we increased the imbalance of error-bearing and error-free windows significantly. For the window size of 3-second in a $\sim$2,5 minutes game, the number of windows with a ghost error was 1 compared to 48 error-free windows.

To address the issue, we applied two new methods to increase, artificially, the number of errors. The first was to increase the number of error-bearing windows via labeling a fixed number of time frames preceding an error with an increasing probability of an error happening. The second was classification, like predicting if an error will happen or not during the next five time frames (for a window size of 3 seconds, that would be in the next 15 seconds). The latter approach proved to work best together with our models and the imbalance was reduced to 15 errors-free to 1 error-bearing window.

B. Model Evaluation Metrics

The errors are unlikely events in our data therefore it is more important to identify one error window (TP) than to identify many error-free windows (TN) in our prediction. Precision tells us what proportion of positive identifications is actually correct (TP/(TP+FP)), and recall explains what proportion of actual positives is identified correctly (TP/(TP+FN)). To be able to consider both metrics at the same time, we decided on the F1 score as the primary metric while comparing the performances of different models. F1 score is the harmonic average of the precision and recall scores. We also considered the area under the ROC curve (AUC) score as a metric well, which explains how well a model distinguishes between classes.

C. Models

We focused our analysis on five models: Logistic Regression (LR), Random Forest (RF), XGBoost (shallow and deep), feed-forward neural network, and a recurrent neural network (LSTM). Acquiring multiple models increased the trust in the relevance of features. One could expect the models to perform similarly well as long as the features are relevant. We used cross-validation with five splits to choose hyper-parameters, such as the number of backward steps (previous windows to consider), the learning rate, the time frame size, the regularization factors, and the recurrent layers' sizes. We concluded our analysis to pick five backward steps (input being a sequence of 5 consecutive time frames), each having a window size of 3 seconds, and using a time frame for our future prediction of 15 seconds.

The logistic regression run for at most 1000 iterations. For the random forest and XGBoost, we considered 300 estimators with a depth of 5. We also considered a shallow variant of the XGBoost, employing only 200 estimators with a depth of 3. These hyperparameters were chosen by a grid search.

The architectures of the neural networks are presented in Fig. 3. The feed-forward was composed of two fully connected layers, first of 128 units, using a leaky rectified linear activation for its hidden units to avoid the activation value of 0.0, and one layer for the final classification using a soft-max activation to output the probability. Both the feedforward and the recurrent model were trained using a categorical cross-entropy loss.

For the recurrent neural network, we employed a stacked long short-term memory (LSTM) architecture. The LSTM units [16] use a gating mechanism to retain or forget data from the previous state and preserve information, solving the vanishing gradient from traditional RNNs, outperforming traditional methods employed in time series analysis [17]. Furthermore, stacking LSTM units had better performance in detecting anomalies (abrupt changes) on multiple datasets [18]. Leaky rectified linear activation for the first layer was used. We found the sizes of 100 and 80 units respectively for the LSTM layers to perform better. Increasing the number of units greatly increased the model's complexity, so we needed to prevent overfitting. We accomplished this by using dropout with a factor of 0.5 before the classification layer and using in both LSTM layers an L1, and L2 regularization [19] with a factor of 1e-5, which performed better than Lasso and Ridge regression alone on our problem.

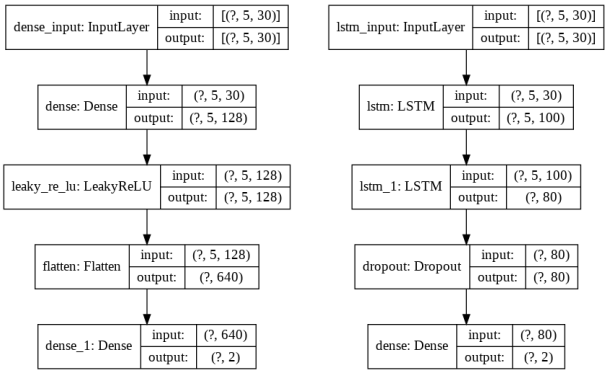

Fig. 3: Model architectures for feed-forward neural network (left) and recurrent neural network (right).

For both models, we ensured the batches have equal numbers of errors and non-errors via oversampling. We used a batch size of 32, the Adam optimizer [20] with a learning rate of 0.001, and we trained the models until the validation loss decreased with the patience of 15 epochs, for the model with the minimum validation loss.

To ensure the correctness of the model, we further performed leave-one-out cross-validation by removing each patient from the training set and analyzed the model's ability to generalize its learning to new patients, as it is critical to ensure it will be able to address patients with different stroke levels or age. A similar approach was applied to leaving a percentage of full games in the validation set to see how the model generalizes new games played. The models were developed by Keras, XGBoost, and Scikit-Learn libraries.

IV. Results

The results are presented in Table I for models trying to predict ghost errors and in Table II for models trying to predict the border-cross errors. The LSTM-based architecture performs better, followed by the feed-forward neural network. We observed that the neural network performs better with lower window sizes since it leads to a bigger dataset for training. Simultaneously, this increases the imbalance, therefore for LR, RF, and XGBoost models, a larger window size led to a better result but not the best results.

Metric	LSTM	Feed Forward	LR	RF	XGBoost	Shallow XGBoost
F1	**0.529**	0.327	0.2644	0.267	0.2198	0.3108
Accuracy	0.917	0.805	0.7334	0.7558	**0.9326**	0.908
Precision	0.432	0.214	0.162	0.167	**0.4754**	0.3098
Recall	0.68	0.689	**0.7194**	0.6702	0.1432	0.312
AUC	**0.807**	0.751	0.727	0.7162	0.566	0.6312

TABLE I: Results for prediction of ghost catch errors

Among two different ways to split the data into training and validation, leave-one-patient-out cross-validation led to an average validation F1-Score of 0.749, which means it

Metric	LSTM	Feed Forward	LR	RF	XGBoost	Shallow XGBoost
F1	**0.844**	0.533	0.5162	0.5182	0.4688	0.5182
Accuracy	**0.781**	0.515	0.6664	0.6868	0.7366	0.7066
Precision	0.885	**0.904**	0.4308	0.4506	0.5296	0.4738
Recall	**0.806**	0.378	0.6446	0.6104	0.421	0.5718
AUC	**0.759**	0.633	0.66	0.663	0.639	0.6648

TABLE II: Results for prediction of wall crashing errors

performs reasonably well on new patients. Another attempt of splitting the dataset by individual games to check the model's ability to learn from a couple of initial games led to an average F1-Score of 0.717.

A. Importance of Features

In order to interpret the results in relation with the extracted features, we computed the permutation importance of the features [21] on the best model and measured the increase in the F1 score after permuting each feature's values. Permutation breaks the tie between the feature and outcome that is, if a feature is important, the score is significantly affected.

Fig. 4 and Fig. 5 present the importance of the features on prediction of the errors. We observed that smoothness (v_ratio, vy_ratio), accuracy (deviance), speed of the motion (vy_mean) and the players affected hand information have highest importance in predicting the border-cross errors. The distance to the center of the map is also important in both error predictions. When the players are moving towards the sides of the map, they tend to do more errors. Possible reasons behind it might be hardness in controlling the arm movements when it is extended or flexed and overshooting the targets which are mostly (4 out of 6 apples) close to the corners of the maps. Similarly, ghost errors might occur when they are far from the center. We observed that features related to the position of the both ghosts (nearest_ghost_diff, nearest_ghost_diff_y, farthest_ghost_diff_x) become more prominent in ghost error prediction. An interesting observation is that accuracy (deviance) and smoothness related metrics have higher importance in predicting ghost catch errors while velocity does not play more crucial role. This might be due to the velocity adjustments of ghosts according to the impairment level of the patients. On the other hand, since ghost avoidance needs more planning and cognitive load, non-smooth and less accurate motion might require more attention towards the motion and the patient might not be attentive enough for the ghosts. Further investigation is needed to understand the relationship between in game motor performance, cognitive load and the motor impairment level of the patients.

V. Discussion

LSTMs have been known for their capability of high-quality feature extraction from time-series data [22]. Their architecture allows them to capture the trends and characteristics of sequences. Since most of our features are derived from sequence data it is expected to observe that the LSTM model performs better with higher F1 scores than other

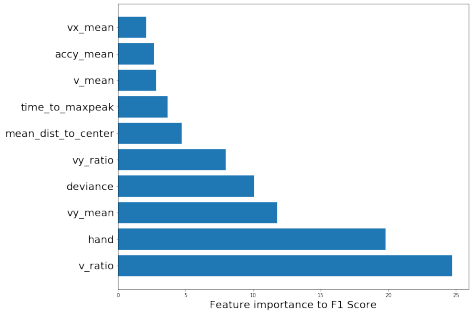

Fig. 4: Features with high importance when predicting border-cross errors

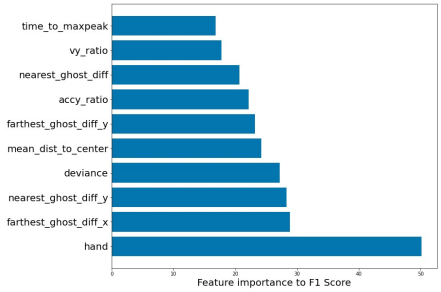

Fig. 5: Features with high importance when predicting ghost catch errors

models which do not consider the periodicity and inner correlation of the time series data.

Since the number and the speed of the ghosts were adapted by the doctor to the impairment level of the patient to avoid frustration, especially ghost catch errors become rare events. On the other hand, since the size of the pathways are same across the all maps, there are more occurrences of wall crashing events. Therefore, the data of cross-border errors is less imbalanced than the ghost-catching errors, and accordingly, the prediction scores of the models are higher in cross-border errors compared to the ghost catch errors. Despite our efforts there is still a need for further approaches focusing on skewed datasets. Using the models in real-time to predict the errors earlier, and then engineering harder configurations that causes more errors may be considered in the future for collecting more balanced datasets.

Another inherent challenge of the dataset is the variation of motion performances of stroke patients. The stroke patients' movements do not have similar means, have very high variance, and come from distinct movement distributions based on their impairment levels. These make it harder for models to generalize the game-play features and detect the errors with a limited number of patients. One can intuitively assume that the models perform better with more data.

VI. Conclusion

In this research effort, we modeled the error prediction in a game as a predictive analysis problem and explored the ability of 5 machine learning models' early prediction of two different types of errors. Our analysis showed promising re-

sults on the usage of an LSTM-based network for predicting motion-accuracy-related errors with an F1 score of 0.844, and for predicting ghost catch errors with an F1 score of 0.529 with AUC (ROC) values 0.759 and 0.807, respectively.

Being able to reliably predict the errors from the game data paves the way for a tailored game design according to the present as well as feature expected performance of each patient. By investigating performance data prior to the error, one can identify the uncontrolled movements of the player that caused the error. Game setup may later be altered to urge or avoid a particular action depending on the treatment need of the patient.

We furthermore adopted an explainable machine learning approach by computing the permutation importance of the features to investigate the relationship between errors with motor performance metrics and game-related features. Results highlight the importance of error types that might be due to motor impairment or cognitive load of the game which needs further investigation.

REFERENCES

[1] E. Flores, G. Tobon, E. Cavallaro, F. I. Cavallaro, J. C. Perry, and T. Keller, "Improving patient motivation in game development for motor deficit rehabilitation," in *Proceedings of the 2008 international conference on advances in computer entertainment technology*, pp. 381–384, 2008.

[2] E. Vergaro, M. Casadio, V. Squeri, P. Giannoni, P. Morasso, and V. Sanguineti, "Self-adaptive robot training of stroke survivors for continuous tracking movements," *Journal of neuroengineering and rehabilitation*, vol. 7, no. 1, pp. 1–12, 2010.

[3] M. Pirovano, R. Mainetti, G. Baud-Bovy, P. L. Lanzi, and N. A. Borghese, "Self-adaptive games for rehabilitation at home," in *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 179–186, IEEE, 2012.

[4] A. Kumar, E. Pirogova, S. S. Mahmoud, and Q. Fang, "Classification of error-related potentials evoked during stroke rehabilitation training," *Journal of Neural Engineering*, vol. 18, no. 5, p. 056022, 2021.

[5] N. Vaughan, B. Gabrys, and V. N. Dubey, "An overview of self-adaptive technologies within virtual reality training," *Computer Science Review*, vol. 22, pp. 65–87, 2016.

[6] D. Johnson and J. Wiles, "Effective affective user interface design in games," *Ergonomics*, vol. 46, no. 13-14, pp. 1332–1345, 2003.

[7] M. Csikszentmihalyi and M. Csikzentmihaly, *Flow: The psychology of optimal experience*, vol. 1990. Harper & Row New York, 1990.

[8] M. A. Guadagnoli and T. D. Lee, "Challenge point: a framework for conceptualizing the effects of various practice conditions in motor learning," *Journal of motor behavior*, vol. 36, no. 2, pp. 212–224.

[9] A. Guneysu Ozgur, M. J. Wessel, W. Johal, K. Sharma, A. Özgür, P. Vuadens, F. Mondada, F. C. Hummel, and P. Dillenbourg, "Iterative design of an upper limb rehabilitation game with tangible robots," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 241–250, 2018.

[10] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining "gamification"," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek '11)*, p. 9–15, 2011.

[11] J. W. Burke, M. McNeill, D. Charles, P. Morrow, J. Crosbie, and S. McDonough, "Serious games for upper limb rehabilitation following stroke," in *IEEE Conference in Games and Virtual Worlds for Serious Applications (VS-GAMES'09)*, pp. 103–110, 2009.

[12] A. Özgür, W. Johal, F. Mondada, and P. Dillenbourg, "Haptic-enabled handheld mobile robots: Design and analysis," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 2449–2461, 2017.

[13] A. Özgür, *Cellulo: Tangible Haptic Swarm Robots for Learning*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2018.

[14] A. G. Ozgur, M. J. Wessel, J. K. Olsen, W. Johal, A. Özgür, F. C. Hummel, and P. Dillenbourg, "Gamified motor training with tangible robots in older adults: a feasibility study and comparison with the young," *Frontiers in aging neuroscience*, vol. 12, 2020.

[15] A. Schwarz, C. M. Kanzler, O. Lambercy, A. R. Luft, and J. M. Veerbeek, "Systematic review on kinematic assessments of upper limb movements after stroke," *Stroke*, vol. 50, no. 3, pp. 718–727, 2019.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 1735–1780, Nov. 1997.

[17] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401, 2018.

[18] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *ESANN*, 2015.

[19] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[21] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," 2019.

[22] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, *et al.*, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, vol. 89, pp. 89–94, 2015.